VIRTUALIZATION FOR THE LHCB EXPERIMENT

E. Bonaccorsi, L. Brarda, M. Chebbi, N. Neufeld, CERN, Geneva, Switzerland
F. Sborzacchi, INFN, Laboratori Nazionali di Frascati, Italy.

Abstract

The LHCb Experiment, one of the four large particle physics detectors at CERN, counts in its Online System more than 2000 servers and embedded systems. As a result of ever-increasing CPU performance in modern servers, many of the applications in the controls system are excellent candidates for virtualization technologies. We see virtualization as an approach to cut down cost, optimize resource usage and manage the complexity of the IT infrastructure of LHCb. Recently we have added a Kernel Virtual Machine (KVM) cluster based on Red Hat Enterprise Virtualization for Servers (RHEV) complementary to the existing Hyper-V cluster devoted only to the virtualization of the windows guests. This paper describes the architecture of our solution based on KVM and RHEV as along with its integration with the existing Hyper-V infrastructure and the Quattor cluster management tools and in particular how we use to run controls applications on a virtualized infrastructure. We present performance results of both the KVM and Hyper-V solutions, problems encountered and a description of the management tools developed for the integration with the Online cluster and LHCb SCADA control system based on PVSS.

INTRODUCTION

LHCb is an experiment set up to explore what happened after the Big Bang that allowed matter to survive and build the Universe we inhabit today, in the specific is a dedicated heavy-flavour physics experiment designed to perform precise measurements of CP violation [1]. The experiment is located at point 8 of the LHC particle accelerator.

The LHCb online system has been designed to run completely isolated and independent, as an autonomous system, it consist of ~2000 physical servers and embedded systems interconnected through 3 main high density routers and ~100 distributions switches.

Hosts are organized in two different local area networks: the Experiment Control System (ECS) [2], illustrated in Figure 1 and the Data Acquisition System (DAQ). The access to the CERN General Purpose Network (GPN) and consequently to internet is provided by Linux and Windows gateways which are secured by a three tier firewall setup.

While the DAQ hosts have been designed to discern the "potentially interesting event" from the huge amount of data produced by the hadrons collisions, zero-suppressed in the front-end electronics [3], the ECS network has been designed to control the experiment, mainly using open sources software wherever possible and the standard LHC SCADA system PVSS in order to control and monitoring high and low voltages, gas and temperatures, etc.

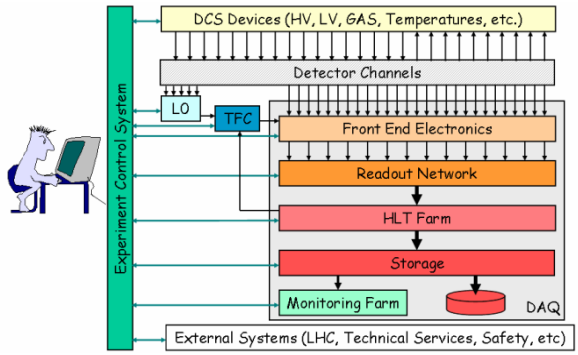

Figure 1: ECS Diagram.

Unlike the DAQ hosts the ECS hosts most of the time underuse resources (with some peak time to time) in terms of memory, network, power, cooling and space.

Taking into account that servers are becoming increasingly powerful, the use of the many-core CPUs accentuates this issue.

Virtualization has in principle a great promise for a control system like ours. It can save power and space and in the long run also money. At the same time it increases the availability and serviceability by abstracting software services from the underlying hardware. However, as we had to learn, the initial investment is rather high and many things have to be taken into account.

The LHCb online team has performed an evaluation of available clustered virtualization implementations focusing mainly on the free edition of Microsoft core Hyper-V [4].

The first part of the project was focused on the virtualization of the public web services and the essential infrastructure services summarized in Table 1.

In this paper we describe further work were we had add a virtualization implementation based on Linux KVM and Red Hat Enterprise Virtualization (RHEV).

Our plan is to migrate from the Microsoft Hyper-V infrastructure to the RHEV infrastructure as well as the deployment of virtual "experiment control PCs", in charge of controlling the detector hardware.

Table 1: Virtualized Systems

Category	Virtualized Systems
Public available	Web Services
Common infrastructure	Firewall, DNS, Domain Controllers, Cron system, DHCP
ECS	Control PCs
Test systems	Dedicated control PCs for testing software and procedures

HYPERVISORS

The hypervisor, also called virtual machine monitor, is a virtualization platform that allows multiple operating systems to run on a single host at the same time.

In the initial part of the study Microsoft Hyper-V had been chosen as hypervisor mainly because the virtualization technology offered by Red Hat/Scientific Linux was in a transition state from XEN to KVM.

A three nodes RHEV cluster has been tested and put into production in parallel with the Hyper-V cluster. The RHEV architecture is illustrated in Figure 2.

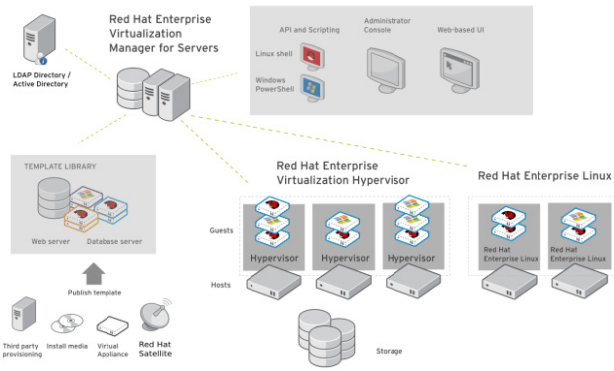

Figure 2: RHEV Architecture.

HARDWARE

The RHEV implementation has been deployed on the same hardware used for the first one and upgraded in terms of memory. The specifications about the memory and the I/O cards are summarised in Table 2.

Table 2: Hardware Specifications

CPU	2 x E5530 @ 2.4 GHz (8 real cores + Hyper Threading)
Memory	6 x 8 GB = 48 GB RAM
Network adapters	2 x 10 Gb network interfaces (for VLAN sharing, 1 linked to ECS) 2 X 1 Gb network interfaces (1 linked to CERN GPN, 1 used for cluster communications)
Fibre channel adapters	2 X 8 Gb Fiber channel switches (linked to two isolated fabrics)

STORAGE AREA NETWORK (SAN)

Virtual disks are stored on a DDN 9900 shared storage system as logical volumes (LV) interconnected through a redundant multipath fibre channel.

The DDN 9900 storage controllers can reach a high level of throughput using a proprietary RAID level called "Direct RAID": each RAID set consist of 10 spindles (disks), of which 8 are used for the data and 2 for the parity.

Three SATA RAID sets ("tiers") and one small SSD RAID set have been exported to the RHEV cluster and configured as a single Volume Group (VG).

The preferable block size for the Logical Units (LUN) in the RAID set is a multiple of 4 Kilobyte (512 Bytes times 8 disks), but unfortunately this value is not supported either by both Hyper-V nor by RHEV, which force us to use a LUN block size of 512 Bytes with the consequential lost of performance in terms of bandwidth and random Input Output Operation per Seconds (IOPS)

STORAGE IOPS

The reason why the DDN9900 is very fast in terms of throughput is the simultaneous access to all the disks in the same RAID set. While this makes the storage extraordinarily fast for sequential I/O operations, the access to all disks at the same time drastically reduces the number of random IOPS.

Considering that each SATA RAID set can perform ~200 random IOPS and that a virtual machine (VM) for standard operations needs at least an average of 30 IOPS, the current storage implementation limit the maximum number of VMs to ~50 using 4 RAID-sets. The effect of adding IOPS is clearly visible in Figure 3.

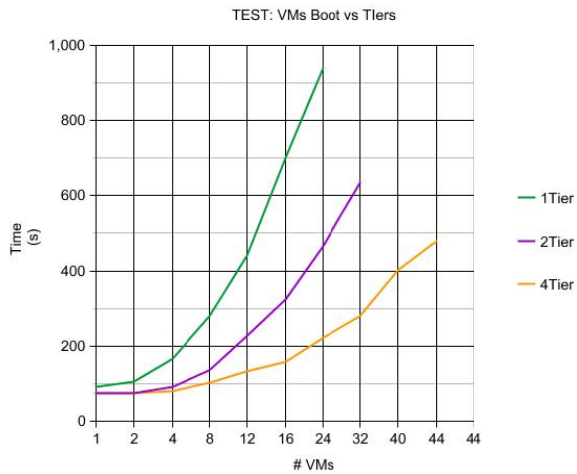

Figure 3: shows how increasing the number of RAID sets (IOPS) improves performances measured in "boot time".

TUNING

Even though there is a fundamental limitation in our current storage hardware a lot of improvement can be obtained by carefully tuning various parameters described in the following.

The main improvements, after having increased the number of RAID sets to the maximum available, have been achieved by switching the VMs default scheduler to NOOP:

By default Red Hat/Scientific Linux uses the CFQ [5] scheduler configured to balance the IO request and it aggregate them to a smallest number of large requests. While the idea of adding "intelligence" to the scheduling of the IO requests is great for a real PC, in a virtual machine this kind of scheduling will just add an additional delay since the smart scheduling will be done

twice: one time by the virtual machine and one time by the hypervisor in which the virtual machine is running.

Significant improvements have been measured as well adding to the default mount options of the VMs filesystems the "noatime,nodiratime" option as well as mounting the /tmp as tmpfs. This last measure keeps temporary files on a local RAM disk rather than soliciting external storage.

By default in ext3 for each read-request a write request will be triggered in order to update the file and directory access time.

Starting a "Name Server/LDAP" caching daemon and disabling IPv6 improved the users experience for the kind of virtual machines which are dedicated to general log-in. In this way the virtual machine does not need to do a request to the DNS and LDAP servers every time an hostname, an IP address or a UID/GID needs to be resolved.

For the virtual machines that are creating a lot of IOPS a solution based on moving the ext3 metadata away from the data in a dedicated SSD RAID set has been put in place. The details of this will be described in a forthcoming, dedicated publication.

This solution which is order of magnitude faster in terms of IOPS compared to the SATA RAID set works also for real machines and is achieved through LVM moving the physical extents in which the metadata is stored to an SSD RAID set.

Regarding the tuning of the hypervisors particular attention was given to the fiber channel interfaces in which we decreased the frame size to 512 Bytes allowing a more number of frames and consequentially of IOPS to the storage in the same interval time.

LHCB VIRTUAL NETWORKS

Live migration of VMs is one of the main advantages of having a virtual infrastructure making the machines less vulnerable to HW failures.

This put same constraint on the network configuration and according to common security procedures three virtual firewalls based have been put in place in order to isolate virtual networks and demilitarized zones. These are shared between the real machines using VLAN through a 10 Gb/s link.

The two 1 Gb/s links are dedicated respectively to cluster management communications and as up-link to CERN network/Internet.

For high-availability reasons the LHCb networks have been linked through a 10Gb/s connection per server with a switch uplink the LHCb core router of 20 Gb/s made by two link on two different linecards using the Link Aggregation Control Protocol (LACP).

A logical map of the virtual network is illustrated in Figure 4.

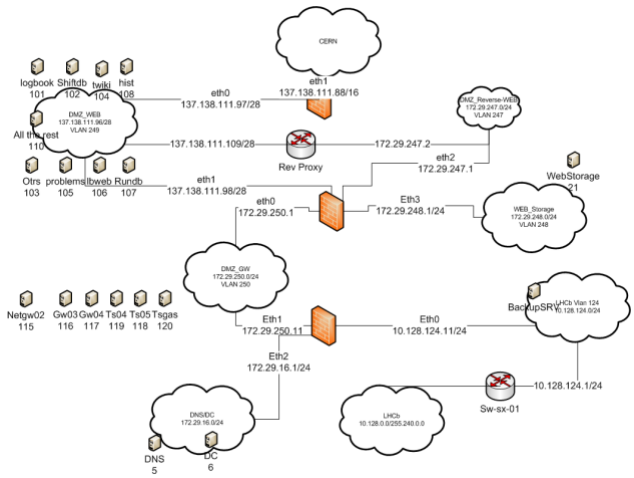

Figure 4: Logical Virtual Networks.

NETWORK PERFORMANCES

We measured the network throughput and the network latency from a KVM and a hyper-v virtual machine with the paravirtualized drivers installed, to a real server inside the LHCb network linked to the core router.

The tests have been done with iperf [6] and ICMP echo requests/replies.

The results are for KVM respectively ~1.50 Gb/s of throughput and ~0.3 ms of latency and for Hyper-V ~900 Mb/s of throughput and ~0.2 ms of latency.

In both hypervisors when the network traffic is filtered and routed by an additional virtual machine the latency time increases to ~0.6 ms and the bandwidth decrease to ~250 Mb/s

INTEGRATION WITH QUATTOR CLUSTER MANAGEMENT TOOL

The main problem in deploying OS on a Hyper-V virtual machine is the lack of pre execution environment (PXE) support when paravirtualized driver are used.

This is not the case for the KVM based VMs because the paravirtualized drivers called VIRTIO are included in the main vanilla kernel since version 2.6.20 and ported back by RedHat/Scientific Linux to version 2.6.18.

The virtual machines are now installed using QUATTOR, a system administration toolkit that provides a powerful, portable, and modular set of tools for the automated installation, configuration, and management of linux clusters and farms, like any real machine in the experiment [7].

ISSUES

Unlike the first study done on Microsoft Hyper-V we did not find any problems with networking, multicast and ACPI, also licensing problems with PVSS are not present. The main problem is the current storage backend, whose RAID system is not optimized for random IOPS.

CONCLUSIONS

The LHCb virtual infrastructure is now based on RHEV. Careful tuning allowed us to achieve very good latency and network and storage throughput.

The current system is however affected by a bottleneck in terms of random IOPS limiting the number of VMs to be executed at the same time.

A new storage solution will be chosen and bought in Q4 2011. The requirements derived from studies conducted for this paper will guarantee a maximum number of concurrent VMs of at least 180.

Once the acquisition will be completed we will continue to deploy more VMs focusing on the control PCs of the experiment.

REFERENCES

[1] LHCb Trigger System TDR, LHCb TDR 10, CERN/LHCC/2003-31, 2003.

[2] An Integrated Experiment Control System, Architecture, and Benefits: The LHCb Approach – IEE Transaction on Nuclear Science, Vol. 51, NO 3, June 2004.

[3] The LHCb Trigger and Data Acquisition System J.-P. Dufey, M. Frank, F. Harris, J. Harvey, B. Jost, P. Mato, H. Mueller.

[4] Virtualization for the LHCb online system – CHEP 2010 – ID 141. E. Bonaccorsi, L. Brarda, G. Moine, N. Neufeld.

[5] http://www.redhat.com/f/pdf/rhel/Oracle-10-g-recommendations-v1_2.pdf.

[6] http://iperf.sourceforge.net/.

[7] https://lbtwiki.cern.ch/bin/view/Online/AdminGuide Quattor.